**Ai documentation**

1.I created two thematic categories of 1-gram words to compare trends in peace-related and conflict related language. I grouped them manually based on their meaning and relevance to the Gaza war context.

**I asked chatgpt**

**“how to creatively group 1-grams and ensure meaningful contrast.**

**Code Developed with AI Help**

**# Define peace and conflict-related 1-gram word groups for comparison.**

**peace\_terms = ['ceasefire', 'resolution', 'security', 'peace']**

**conflict\_terms = ['strike', 'invasion', 'rocket', 'military']**

**all\_terms = peace\_terms + conflict\_terms**

2. I created a new column called "category" to label each 1-gram as either "Peace" or "Conflict" using a conditional check. This allows clearer visualization and comparison of the two themes in later plots.

**I asked ChatGPT**

**“how to label values in a DataFrame column based on predefined list”**

**Code developed with the help of AI**

**# Create a new column to label each 1-gram as either Peace or Conflict**

**df2["category"] = df2["1-gram"].apply(lambda x: "Peace" if x in peace\_terms else "Conflict")**

3. I used plotly.subplots.make\_subplots to manually create a subplot layout that displays separate bar charts for each selected 1-gram (both peace and conflict terms). This allowed me to compare frequencies side-by-side in a custom arrangement

**I asked ChatGPT**

**“I have filtered a list of 1-gram words from my CSV (some about peace, some about conflict). I want to plot related terms separately to see how its frequency changes over the months.**

**This is the code I got**

**# Set up subplots with 1 row and 2 columns (side by side bar graphs)**

**fig = make\_subplots(**

**rows=1, cols=2,**

**subplot\_titles=("Peace Terms", "Conflict Terms"), #titles for subplots**

**shared\_yaxes=True # share the same y-axis scale for better comparison**

**)**